



Texas A&M University - Commerce  
Department of Computer Science

# Predictive Precision: Dissecting SVM and Random Forests in Heart Disease Risk Analysis

Lakshmi Chandana Narra

*Supervisor:* Derek Harter, Ph.D.

A report submitted in partial fulfilment of the requirements of  
Texas A&M University - Commerce for the degree of  
Master of Science in *Computer Science*

February 3, 2024

## Declaration

I, Lakshmi Chandana Narra, of the Department of Computer Science, Texas A&M University - Commerce, confirm that this is my own work and figures, tables, equations, code snippets, artworks, and illustrations in this report are original and have not been taken from any other person's work, except where the works of others have been explicitly acknowledged, quoted, and referenced. I understand that if failing to do so will be considered a case of plagiarism. Plagiarism is a form of academic misconduct and will be penalised accordingly.

I give consent to a copy of my report being shared with future students as an exemplar.

I give consent for my work to be made available more widely to members of TAMUC and public with interest in teaching, learning and research.

Lakshmi Chandana Narra  
February 3, 2024

## Abstract

In recent times, the global rise in cardiovascular diseases has become increasingly prevalent, influenced by evolving lifestyles and societal factors. The critical need for timely detection and continual monitoring is accentuated, particularly in regions with limited access to medical resources. This study addresses these challenges by developing predictive models for heart disease, specifically investigating the performance of Support Vector Machines (SVM) and Random Forest algorithms.

Utilizing a public health dataset on patient heart health, including information from medical procedures and ongoing patient monitoring, the research uniquely centers on the comparative analysis of SVM and Random Forests. In contrast to broader discussions found in existing literature, this study exclusively delves into the intricacies of these two machine learning algorithms. The assessment employs key accuracy metrics, such as the confusion matrix, Accuracy, precision, and F1 score, to comprehensively evaluate and compare their predictive capabilities.

The study reveals the strengths and weaknesses inherent in SVM and Random Forests, offering valuable insights for the development of effective prediction models tailored to the challenges of cardiovascular health. Focused on these two algorithms, the research aligns with the evolving landscape of machine learning in healthcare, presenting a concentrated perspective on their potential contributions.

In summary, this research makes a significant contribution by addressing the pressing need for accurate predictive models in cardiovascular health, placing emphasis on the comparative analysis of SVM and Random Forest algorithms. The findings seek to inform healthcare strategies, providing essential insights for the development of efficient and focused prediction models in the context of heart disease

**Keywords:** Machine Learning, Random Forest, Support Vector Machine, Cardiovascular disease , Healthcare

## **Acknowledgements**

An acknowledgements section is optional. You may like to acknowledge the support and help of your supervisor(s), friends, or any other person(s), department(s), institute(s), etc. If you have been provided specific facility from department/school acknowledged so.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Background . . . . .	1
1.2	Problem statement . . . . .	1
1.3	Aims and objectives . . . . .	2
1.4	Solution approach . . . . .	2
1.4.1	A subsection 1 . . . . .	2
1.4.2	A subsection 2 . . . . .	3
1.5	Summary of contributions and achievements . . . . .	3
1.6	Organization of the report . . . . .	3
<b>2</b>	<b>Literature Review</b>	<b>4</b>
2.1	Example of in-text citation of references in $\text{\LaTeX}$ . . . . .	4
2.2	Example of “risk” of unintentional plagiarism . . . . .	5
2.3	Critique of the review . . . . .	5
2.4	Summary . . . . .	5
<b>3</b>	<b>Methodology</b>	<b>6</b>
3.1	Examples of the sections of a methodology chapter . . . . .	6
3.1.1	Example of a software/Web development main text structure . . . . .	6
3.1.2	Example of an algorithm analysis main text structure . . . . .	6
3.1.3	Example of an application type main text structure . . . . .	6
3.1.4	Example of a science lab-type main text structure . . . . .	7
3.2	Example of an Equation in $\text{\LaTeX}$ . . . . .	9
3.3	Example of a Figure in $\text{\LaTeX}$ . . . . .	9
3.4	Example of an algorithm in $\text{\LaTeX}$ . . . . .	10
3.5	Example of code snippet in $\text{\LaTeX}$ . . . . .	10
3.6	Example of in-text citation style . . . . .	12
3.6.1	Example of the equations and illustrations placement and reference in the text . . . . .	12
3.6.2	Example of the equations and illustrations style . . . . .	12
3.7	Summary . . . . .	13
<b>4</b>	<b>Results</b>	<b>14</b>
4.1	A section . . . . .	14
4.2	Example of a Table in $\text{\LaTeX}$ . . . . .	15

<i>CONTENTS</i>		v
4.3	Example of captions style . . . . .	15
4.4	Summary . . . . .	15
<b>5</b>	<b>Discussion and Analysis</b>	<b>16</b>
5.1	A section . . . . .	16
5.2	Significance of the findings . . . . .	16
5.3	Limitations . . . . .	16
5.4	Summary . . . . .	16
<b>6</b>	<b>Conclusions and Future Work</b>	<b>17</b>
6.1	Conclusions . . . . .	17
6.2	Future work . . . . .	17
<b>7</b>	<b>Reflection</b>	<b>18</b>
	<b>Appendices</b>	<b>20</b>
<b>A</b>	<b>An Appendix Chapter (Optional)</b>	<b>20</b>
<b>B</b>	<b>An Appendix Chapter (Optional)</b>	<b>21</b>

# List of Figures

3.1	Example figure in $\text{\LaTeX}$ . . . . .	9
-----	---	---

# List of Tables

3.1	Undergraduate report template structure . . . . .	7
3.2	Example of a software engineering-type report structure . . . . .	7
3.3	Example of an algorithm analysis type report structure . . . . .	8
3.4	Example of an application type report structure . . . . .	8
3.5	Example of a science lab experiment-type report structure . . . . .	8
4.1	Example of a table in $\LaTeX$ . . . . .	15



# List of Abbreviations

SMPCS      School of Mathematical, Physical and Computational Sciences

# Chapter 1

## Introduction

**Investigated Problem:** Cardiovascular diseases represent a formidable global health challenge, with their prevalence escalating and ranking among the primary causes of morbidity and mortality. The pressing concern is the need for robust predictive models to address the increasing burden of heart diseases, enabling early detection and effective risk mitigation strategies.

**Scope and Context of the Project:** This research project delves into the development and evaluation of predictive models for heart disease using machine learning algorithms, specifically focusing on Support Vector Machines (SVM) and Random Forest. The scope encompasses a comprehensive analysis of these algorithms, exploring their capabilities and limitations in accurately predicting the risk of cardiovascular events. The context of the project revolves around leveraging a diverse dataset derived from various medical procedures and continuous patient monitoring to enhance our understanding of heart disease prediction.

### 1.1 Background

The background of this study lies in the evolving landscape of lifestyle, dietary habits, and healthcare dynamics that contribute to the increasing prevalence of cardiovascular diseases. The significance of early detection and continuous monitoring underscores the importance of advanced predictive modeling techniques. Against this backdrop, the research aims to contribute to the field of cardiovascular health by providing insights into the efficacy of SVM and Random Forest algorithms.

In summary, the investigated problem centers on the escalating prevalence of cardiovascular diseases, and the project's scope involves the development and evaluation of predictive models using SVM and Random Forest. The background highlights the contextual relevance of advanced predictive modeling in addressing the challenges posed by heart diseases in the contemporary healthcare landscape.

### 1.2 Problem statement

Globally, the prevalence of cardiovascular diseases is rising, which emphasizes the need for efficient predictive models to identify risks early. The purpose of this research is to compare the prediction

powers of Random Forest and Support Vector Machines (SVM) for forecasting the risk of heart disease. The study places a strong emphasis on evaluating efficiency and accuracy indicators, including the F1 score, accuracy, precision, and confusion matrix.

To guarantee the research's viability, comprehensive assessment of heart disease datasets will be utilized. This study holds significance because it has a particular emphasis on cardiovascular health and aims to provide insightful information to the field. The study aims to fulfill the urgent need for sophisticated predictive models in the field of heart disease by providing answers within the stipulated time period.

### 1.3 Aims and objectives

#### **Aims:**

This research project's main goal is to evaluate and contrast the effectiveness of Random Forest and Support Vector Machines (SVM) algorithms in relation to risk assessments for heart disease. The main objective is to advance predictive modeling methods for accurate assessment and early identification of cardiac disease.

#### **Objectives:**

The Kaggle online dataset resource website is the source of the heart disease dataset that will be analyzed for this research project. Utilize Random Forest and Support Vector Machine (SVM) algorithms to classify data related to heart illness, investigating the potential of each method. Examine and contrast the SVM and Random Forest prediction performances on the chosen datasets related to heart disease in order to pinpoint their advantages and disadvantages.

Methodically adjust multiple variables for both SVM and Random Forest models to investigate differences in classification results. Choose the SVM and Random Forest parameter setups that produce the most accurate and dependable classification results. Finding the best parameter values for the best prediction results is the key objective.

### 1.4 Solution approach

Briefly describe the solution approach and the methodology applied in solving the set aims and objectives.

Depending on the project, you may like to alter the "heading" of this section. Check with your supervisor. Also, check what subsection or any other section that can be added in or removed from this template.

#### 1.4.1 A subsection 1

You may or may not need subsections here. Depending on your project's needs, add two or more subsection(s). A section takes at least two subsections.

### 1.4.2 A subsection 2

Depending on your project's needs, add more section(s) and subsection(s).

#### A subsection 1 of a subsection

The command `\subsubsection{}` creates a paragraph heading in  $\text{\LaTeX}$ .

#### A subsection 2 of a subsection

Write your text here...

## 1.5 Summary of contributions and achievements

Describe clearly what you have done/created/achieved and what the major results and their implications are.

## 1.6 Organization of the report

Describe the outline of the rest of the report here. Let the reader know what to expect ahead in the report. Describe how you have organized your report.

**Example: how to refer a chapter, section, subsection.** This report is organised into seven chapters. Chapter 2 details the literature review of this project. In Section 3...

**Note:** Take care of the word like "Chapter," "Section," "Figure" etc. before the  $\text{\LaTeX}$  command `\ref{}`. Otherwise, a sentence will be confusing. For example, In 2 literature review is described. In this sentence, the word "Chapter" is missing. Therefore, a reader would not know whether 2 is for a Chapter or a Section or a Figure.

## Chapter 2

# Literature Review

A literature review chapter can be organized in a few sections with appropriate titles. A literature review chapter might contain the following:

1. A review of the state-of-the-art (include theories and solutions) of the field of research.
2. A description of the project in the context of existing literature and products/systems.
3. An analysis of how the review is relevant to the intended application/system/problem.
4. A critique of existing work compared with the intended work.

Note that your literature review should demonstrate the significance of the project.

### 2.1 Example of in-text citation of references in $\LaTeX$

The references in a report relate your content with the relevant sources, papers, and the works of others. To include references in a report, we *cite* them in the texts. In MS-Word, EndNote, or MS-Word references, or plain text as a list can be used. Similarly, in  $\LaTeX$ , you can use the “thebibliography” environment, which is similar to the plain text as a list arrangement like the MS word. However, In  $\LaTeX$ , the most convenient way is to use the BibTex, which takes the references in a particular format [see references.bib file of this template] and lists them in style [APA, Harvard, etc.] as we want with the help of proper packages.

These are the examples of how to *cite* external sources, seminal works, and research papers. In  $\LaTeX$ , if you use “**BibTex**” you do not have to worry much since the proper use of a bibliography style package like “agsm for the Harvard style” and little rectification of the content in a BiBText source file [In this template, BibTex are stored in the “references.bib” file], we can conveniently generate a reference style.

Take a note of the commands `\cite{}` and `\citep{}`. The command `\cite{}` will write like “Author et al. (2019)” style for Harvard, APA and Chicago style. The command `\citep{}` will write like “(Author et al., 2019).” Depending on how you construct a sentence, you need to use them smartly. Check the examples of **in-text citation** of sources listed here [This template recommends the **Harvard style** of referencing.]:

- ? has written a comprehensive guide on writing in  $\LaTeX$  [Example of `\cite{}` ].

- If  $\LaTeX$  is used efficiently and effectively, it helps in writing a very high-quality project report (?) [Example of `\citep{}` ].
- A detailed APA, Harvard, and Chicago referencing style guide are available in (David Lapp, 1988).

Example of a numbered list:

1. ? has written a comprehensive guide on writing in  $\LaTeX$ .
2. If  $\LaTeX$  is used efficiently and effectively, it helps in writing a very high-quality project report (?)

## 2.2 Example of “risk” of unintentional plagiarism

Using other sources, ideas, and material always bring with it a risk of unintentional plagiarism.

**MUST:** do read the university guidelines on the definition of plagiarism as well as the guidelines on how to avoid plagiarism (University of Reading, 2019).

## 2.3 Critique of the review

Describe your main findings and evaluation of the literature.

## 2.4 Summary

Write a summary of this chapter

## Chapter 3

# Methodology

We mentioned in Chapter 1 that a project report's structure could follow a particular paradigm. Hence, the organization of a report (effectively the Table of Content of a report) can vary depending on the type of project you are doing. Check which of the given examples suit your project. Alternatively, follow your supervisor's advice.

### 3.1 Examples of the sections of a methodology chapter

A general report structure is summarised (suggested) in Table 3.1. Table 3.1 describes that, in general, a typical report structure has three main parts: (1) front matter, (2) main text, and (3) end matter. The structure of the front matter and end matter will remain the same for all the undergraduate final year project report. However, the main text varies as per the project's needs.

#### 3.1.1 Example of a software/Web development main text structure

Notice that the “methodology” Chapter of Software/Web development in Table 3.2 takes a standard software engineering paradigm (approach). Alternatively, these suggested sections can be the chapters of their own. Also, notice that “Chapter 5” in Table 3.2 is “Testing and Validation” which is different from the general report template mentioned in Table 3.1. Check with your supervisor if in doubt.

#### 3.1.2 Example of an algorithm analysis main text structure

Some project might involve the implementation of a state-of-the-art algorithm and its performance analysis and comparison with other algorithms. In that case, the suggestion in Table 3.3 may suit you the best.

#### 3.1.3 Example of an application type main text structure

If you are applying some algorithms/tools/technologies on some problems/datasets/etc., you may use the methodology section prescribed in Table 3.4.

Table 3.1: Undergraduate report template structure

Frontmatter	Title Page
	Abstract
	Acknowledgements
	Table of Contents
	List of Figures
	List of Tables
	List of Abbreviations
Main text	Chapter 1 Introduction
	Chapter 2 Literature Review
	Chapter 3 Methodology
	Chapter 4 Results
	Chapter 5 Discussion and Analysis
	Chapter 6 Conclusions and Future Work
	Chapter 7 Refection
End matter	References
	Appendices (Optional)
	Index (Optional)

Table 3.2: Example of a software engineering-type report structure

Chapter 1	Introduction
Chapter 2	Literature Review
Chapter 3	Methodology
	Requirements specifications
	Analysis
	Design
	Implementations
Chapter 4	Testing and Validation
Chapter 5	Results and Discussion
Chapter 6	Conclusions and Future Work
Chapter 7	Reflection

3.1.4 Example of a science lab-type main text structure

If you are doing a science lab experiment type of project, you may use the methodology section suggested in Table 3.5. In this kind of project, you may refer to the “Methodology” section as “Materials and Methods.”



Table 3.3: Example of an algorithm analysis type report structure

Chapter 1	Introduction	
Chapter 2	Literature Review	
Chapter 3	Methodology	Algorithms descriptions Implementations Experiments design
Chapter 4	Results	
Chapter 5	Discussion and Analysis	
Chapter 6	Conclusion and Future Work	
Chapter 7	Reflection	

Table 3.4: Example of an application type report structure

Chapter 1	Introduction	
Chapter 2	Literature Review	
Chapter 3	Methodology	Problems (tasks) descriptions Algorithms/tools/technologies/etc. descriptions Implementations Experiments design and setup
Chapter 4	Results	
Chapter 5	Discussion and Analysis	
Chapter 6	Conclusion and Future Work	
Chapter 7	Reflection	

Table 3.5: Example of a science lab experiment-type report structure

Chapter 1	Introduction	
Chapter 2	Literature Review	
Chapter 3	Materials and Methods	Problems (tasks) description Materials Procedures Implementations Experiment set-up
Chapter 4	Results	
Chapter 5	Discussion and Analysis	
Chapter 6	Conclusion and Future Work	
Chapter 7	Reflection	

## 3.2 Example of an Equation in $\text{\LaTeX}$

Eq. 3.1 [note that this is an example of an equation’s in-text citation] is an example of an equation in  $\text{\LaTeX}$ . In Eq. (3.1),  $s$  is the mean of elements  $x_i \in \mathbf{x}$ :

$$s = \frac{1}{N} \sum_{i=1}^N x_i. \quad (3.1)$$

Have you noticed that all the variables of the equation are defined using the **in-text** maths command  $\$.$ , and Eq. (3.1) is treated as a part of the sentence with proper punctuation? Always treat an equation or expression as a part of the sentence.

## 3.3 Example of a Figure in $\text{\LaTeX}$

Figure 3.1 is an example of a figure in  $\text{\LaTeX}$ . For more details, check the link:

[wikibooks.org/wiki/LaTeX/Floats,\\_Figures\\_and\\_Captions](http://wikibooks.org/wiki/LaTeX/Floats,_Figures_and_Captions).

Keep your artwork (graphics, figures, illustrations) clean and readable. At least 300dpi is a good resolution of a PNG format artwork. However, an SVG format artwork saved as a PDF will produce the best quality graphics. There are numerous tools out there that can produce vector graphics and let you save that as an SVG file and/or as a PDF file. One example of such a tool is the “Flow algorithm software”. Here is the link for that: [flowgorithm.org](http://flowgorithm.org).

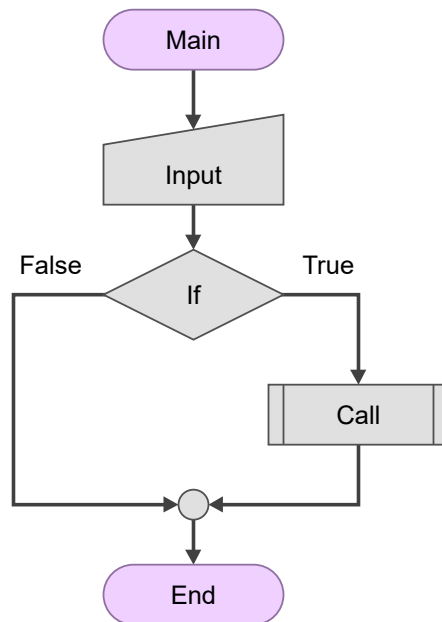


Figure 3.1: Example figure in  $\text{\LaTeX}$ .

### 3.4 Example of an algorithm in $\text{\LaTeX}$

Algorithm 1 is a good example of an algorithm in  $\text{\LaTeX}$ .

---

**Algorithm 1** Example caption: sum of all even numbers

---

**Input:**  $\mathbf{x} = x_1, x_2, \dots, x_N$

**Output:** *EvenSum* (Sum of even numbers in  $\mathbf{x}$ )

```

1: function EVENSUMMATION( $\mathbf{x}$ )
2:   EvenSum  $\leftarrow$  0
3:    $N \leftarrow \text{length}(\mathbf{x})$ 
4:   for  $i \leftarrow 1$  to  $N$  do
5:     if  $x_i \bmod 2 == 0$  then                                ▷ check if a number is even?
6:       EvenSum  $\leftarrow$  EvenSum +  $x_i$ 
7:     end if
8:   end for
9:   return EvenSum
10: end function

```

---

### 3.5 Example of code snippet in $\text{\LaTeX}$

Code Listing 3.1 is a good example of including a code snippet in a report. While using code snippets, take care of the following:

- do not paste your entire code (implementation) or everything you have coded. Add code snippets only.
- The algorithm shown in Algorithm 1 is usually preferred over code snippets in a technical/-scientific report.
- Make sure the entire code snippet or algorithm stays on a single page and does not overflow to another page(s).

Here are three examples of code snippets for three different languages (Python, Java, and CPP) illustrated in Listings 3.1, 3.2, and 3.3 respectively.

```

1 import numpy as np
2
3  $\mathbf{x}$  = [0, 1, 2, 3, 4, 5] # assign values to an array
4 evenSum = evenSummation( $\mathbf{x}$ ) # call a function
5
6 def evenSummation( $\mathbf{x}$ ):
7     evenSum = 0
8      $n = \text{len}(\mathbf{x})$ 
9     for  $i$  in  $\text{range}(n)$ :
10         if  $\text{np.mod}(\mathbf{x}[i], 2) == 0$ : # check if a number is even?
11             evenSum = evenSum +  $\mathbf{x}[i]$ 
12     return evenSum

```

Listing 3.1: Code snippet in  $\text{\LaTeX}$  and this is a Python code example

Here we used the “\clearpage” command and forced-out the second listing example onto the next page.

```

1 public class EvenSum{
2     public static int evenSummation(int[] x){
3         int evenSum = 0;
4         int n = x.length;
5         for(int i = 0; i < n; i++){
6             if(x[i]%2 == 0){ // check if a number is even?
7                 evenSum = evenSum + x[i];
8             }
9         }
10        return evenSum;
11    }
12    public static void main(String[] args){
13        int[] x = {0, 1, 2, 3, 4, 5}; // assign values to an array
14        int evenSum = evenSummation(x);
15        System.out.println(evenSum);
16    }
17 }

```

Listing 3.2: Code snippet in  $\LaTeX$  and this is a Java code example

```

1 int evenSummation(int x[]){
2     int evenSum = 0;
3     int n = sizeof(x);
4     for(int i = 0; i < n; i++){
5         if(x[i]%2 == 0){ // check if a number is even?
6             evenSum = evenSum + x[i];
7         }
8     }
9     return evenSum;
10 }
11
12 int main(){
13     int x[] = {0, 1, 2, 3, 4, 5}; // assign values to an array
14     int evenSum = evenSummation(x);
15     cout<<evenSum;
16     return 0;
17 }

```

Listing 3.3: Code snippet in  $\LaTeX$  and this is a C/C++ code example

## 3.6 Example of in-text citation style

### 3.6.1 Example of the equations and illustrations placement and reference in the text

Make sure whenever you refer to the equations, tables, figures, algorithms, and listings for the first time, they also appear (placed) somewhere on the same page or in the following page(s). Always make sure to refer to the equations, tables and figures used in the report. Do not leave them without an **in-text citation**. You can refer to equations, tables and figures more than once.

### 3.6.2 Example of the equations and illustrations style

Write **Eq.** with an uppercase “Eq” for an equation before using an equation number with (`\eqref{.}`). Use “Table” to refer to a table, “Figure” to refer to a figure, “Algorithm” to

refer to an algorithm and “Listing” to refer to listings (code snippets). Note that, we do not use the articles “a,” “an,” and “the” before the words Eq., Figure, Table, and Listing, but you may use an article for referring the words figure, table, etc. in general.

For example, the sentence “A report structure is shown in **the** Table 3.1” should be written as “A report structure is shown **in** Table 3.1.”

## 3.7 Summary

Write a summary of this chapter.

**Note:** In the case of **software engineering** project a Chapter “**Testing and Validation**” should precede the “Results” chapter. See Section 3.1.1 for report organization of such project.

## Chapter 4

# Results

The results chapter tells a reader about your findings based on the methodology you have used to solve the investigated problem. For example:

- If your project aims to develop a software/web application, the results may be the developed software/system/performance of the system, etc., obtained using a relevant methodological approach in software engineering.
- If your project aims to implement an algorithm for its analysis, the results may be the performance of the algorithm obtained using a relevant experiment design.
- If your project aims to solve some problems/research questions over a collected dataset, the results may be the findings obtained using the applied tools/algorithms/etc.

Arrange your results and findings in a logical sequence.

### 4.1 A section

...

## 4.2 Example of a Table in $\text{\LaTeX}$

Table 4.1 is an example of a table created using the package  $\text{\LaTeX}$  “booktabs.” do check the link: [wikibooks.org/wiki/LaTeX/Tables](http://wikibooks.org/wiki/LaTeX/Tables) for more details. A table should be clean and readable. Unnecessary horizontal lines and vertical lines in tables make them unreadable and messy. The example in Table 4.1 uses a minimum number of lines (only necessary ones). Make sure that the top rule and bottom rule (top and bottom horizontal lines) of a table are present.

Table 4.1: Example of a table in  $\text{\LaTeX}$

Bike		
Type	Color	Price (£)
Electric	black	700
Hybrid	blue	500
Road	blue	300
Mountain	red	300
Folding	black	500

## 4.3 Example of captions style

- The **caption of a Figure (artwork)** goes **below** the artwork (Figure/Graphics/illustration). See example artwork in Figure 3.1.
- The **caption of a Table** goes **above** the table. See the example in Table 4.1.
- The **caption of an Algorithm** goes **above** the algorithm. See the example in Algorithm 1.
- The **caption of a Listing** goes **below** the Listing (Code snippet). See example listing in Listing 3.1.

## 4.4 Summary

Write a summary of this chapter.



## **Chapter 5**

# **Discussion and Analysis**

Depending on the type of project you are doing, this chapter can be merged with “Results” Chapter as “ Results and Discussion” as suggested by your supervisor.

In the case of software development and the standalone applications, describe the significance of the obtained results/performance of the system.

### **5.1 A section**

Discussion and analysis chapter evaluates and analyses the results. It interprets the obtained results.

### **5.2 Significance of the findings**

In this chapter, you should also try to discuss the significance of the results and key findings, in order to enhance the reader’s understanding of the investigated problem

### **5.3 Limitations**

Discuss the key limitations and potential implications or improvements of the findings.

### **5.4 Summary**

Write a summary of this chapter.

## Chapter 6

# Conclusions and Future Work

### 6.1 Conclusions

Typically a conclusions chapter first summarizes the investigated problem and its aims and objectives. It summarizes the critical/significant/major findings/results about the aims and objectives that have been obtained by applying the key methods/implementations/experiment set-ups. A conclusions chapter draws a picture/outline of your project's central and the most significant contributions and achievements.

A good conclusions summary could be approximately 300–500 words long, but this is just a recommendation.

A conclusions chapter followed by an abstract is the last things you write in your project report.

### 6.2 Future work

This section should refer to Chapter 4 where the author has reflected their criticality about their own solution. The future work is then sensibly proposed in this section.

**Guidance on writing future work:** While working on a project, you gain experience and learn the potential of your project and its future works. Discuss the future work of the project in technical terms. This has to be based on what has not been yet achieved in comparison to what you had initially planned and what you have learned from the project. Describe to a reader what future work(s) can be started from the things you have completed. This includes identifying what has not been achieved and what could be achieved.

A good future work summary could be approximately 300–500 words long, but this is just a recommendation.

## Chapter 7

# Reflection

Write a short paragraph on the substantial learning experience. This can include your decision-making approach in problem-solving.

**Some hints:** You obviously learned how to use different programming languages, write reports in  $\text{\LaTeX}$  and use other technical tools. In this section, we are more interested in what you thought about the experience. Take some time to think and reflect on your individual project as an experience, rather than just a list of technical skills and knowledge. You may describe things you have learned from the research approach and strategy, the process of identifying and solving a problem, the process research inquiry, and the understanding of the impact of the project on your learning experience and future work.

Also think in terms of:

- what knowledge and skills you have developed
- what challenges you faced, but was not able to overcome
- what you could do this project differently if the same or similar problem would come
- rationalize the divisions from your initial planned aims and objectives.

A good reflective summary could be approximately 300–500 words long, but this is just a recommendation.

**Note:** The next chapter is “**References**,” which will be automatically generated if you are using BibTeX referencing method. This template uses BibTeX referencing. Also, note that there is difference between “References” and “Bibliography.” The list of “References” strictly only contain the list of articles, paper, and content you have cited (i.e., refereed) in the report. Whereas Bibliography is a list that contains the list of articles, paper, and content you have cited in the report plus the list of articles, paper, and content you have read in order to gain knowledge from. We recommend to use only the list of “References.”

# References

David Lapp (1988), 'Public health dataset for heart disease prediction'.

**URL:** <https://www.kaggle.com/datasets/johnsmith88/heart-disease-dataset>

University of Reading (2019), 'Avoiding unintentional plagiarism: Guidance on citing references for students at the university of reading: Styles of referencing'. (accessed October 26, 2019).

**URL:** <https://libguides.reading.ac.uk/citing-references/avoidingplagiarism>

## Appendix A

### An Appendix Chapter (Optional)

Some lengthy tables, codes, raw data, length proofs, etc. which are **very important but not essential part** of the project report goes into an Appendix. An appendix is something a reader would consult if he/she needs extra information and a more comprehensive understating of the report. Also, note that you should use one appendix for one idea.

An appendix is optional. If you feel you do not need to include an appendix in your report, avoid including it. Sometime including irrelevant and unnecessary materials in the Appendices may unreasonably increase the total number of pages in your report and distract the reader.

## **Appendix B**

### **An Appendix Chapter (Optional)**

...